CSE 493 G1/ 599 G1
Deep Learning
Autumn 2024 Quiz 2

SOLUTIONS

October 25, 2024

Full Name: _____

UW Net ID: _____

Question		Score
True/False	(4 pts)	
Multiple Choice	(8 pts)	
Short Answer	(8 pts)	
Total	(20 pts)	

Welcome to the CSE 493 G1 Quiz 2!

- The exam is 30 min and is **double-sided**.
- No electronic devices are allowed.

I agree to uphold the University of Washington Student Conduct Code during this exam.

Signature: _____ Date: _____

Good luck!

1 True / False (4 points) - Recommended 4 Minutes

Fill in the circle next to True or False, or fill in neither. Fill it in completely like this: ●. No explanations are required.

Scoring: Correct answer is worth 1 points.

1.1 You know you have a good training regime if there is no gap between your train accuracy and val accuracy.

◯ True

◯ False

SOLUTION:
False. While a small gap can be desirable, having no gap could also indicate underfitting, where the model is too simple to learn the underlying patterns in the data.

1.2 One of the reasons that ReLU is better than Sigmoid is that its outputs are zero-centered.

◯ True

◯ False

SOLUTION:
False. ReLU outputs are not zero-centered as they are always non-negative (outputs are in $[0, \infty)$). In fact, neither ReLU nor Sigmoid is zero-centered. ReLU's advantages over Sigmoid include preventing the vanishing gradient problem and computational efficiency, but zero-centered outputs is not one of them.

1.3 After applying max pooling with spatial extent 2, stride 2, and zero padding on an input with $W \times H \times C$, (where W, C, H are all divisible by 2), the number of values in the output shape is one eighth that of the input.

◯ True

◯ False

SOLUTION:
False. Max pooling with spatial extent 2 and stride 2 reduces both width and height by a factor of 2, while keeping the number of channels constant. Therefore, the output shape will be (W/2) × (H/2) × C, which means the number of values in the output is one fourth (not one eighth) of the input.

1.4 Saliency maps compute the gradient of class scores with respect to image pixels to understand which pixels are important for the final prediction of the model.

◯ True

◯ False

SOLUTION:
True. The statement accurately describes both the computation method and purpose of saliency maps.

2 Multiple Choices (8 points) - Recommended 8 Minutes

Fill in the circle next to the letter(s) of your choice (like this: ●). No explanations are required. **Choose ALL options that apply.**

Each question is worth 4 points and the answer may contain one or more options. Selecting all of the correct options and none of the incorrect options will get full credits. For questions with multiple correct options, each incorrect or missing selection gets a 2-point deduction (up to 4 points).

2.1 Let A_1 and A_2 be two activation maps both of size $W \times H \times C$. They correspond to two elements in a batch of size 2. Let $2 \cdot A_1 = A_2$. Let B_1, B_2 correspond to A_1, A_2 after applying some normalization N. Which of the following are always true regardless of the value of A_1 and A_2?

◯ A: If N is layer norm, then $B_1 = B_2$

◯ B: If N is batch norm, then $B_1 = B_2$

◯ C: If N is instance norm, then $B_1 = B_2$

◯ D: The $L2$ norm of the elements in B_1 will be smaller than that of A_1 if N is layer norm.

SOLUTION:

◯ A: True. Since layer norm normalizes each sample independently using μ_1, σ_1 for A_1 and $\mu_2 =_2 \mu_1, \sigma_2 =_2 \sigma_1$ for $A_2 =_2 A_1$, we have:

$$B_1 = (A_1 - \mu_1)/\sigma_1$$

$$B_2 = (2A_1 - 2\mu_1)/(2\sigma_1) = (A_1 - \mu_1)/\sigma_1$$

Therefore $B_1 = B_2$

◯ B: False. Batch norm normalizes across the batch dimension, so when one value is twice the other, the normalized values will not generally be equal. The scaling relationship is not preserved after normalization.

◯ C: True. Similar to layer norm, instance norm processes each sample independently but per channel:

$$B_1 = (A_1 - \mu_1)/\sigma_1$$

$$B_2 = (2A_1 - 2\mu_1)/(2\sigma_1) = (A_1 - \mu_1)/\sigma_1$$

Therefore $B_1 = B_2$

◯ D: False. Layer normalization standardizes the distribution to have unit variance, but this does not guarantee that the $L2$ norm of the normalized values will be smaller than the original. This depends on the specific values in A_1.

2.2 Which of the following is true about optimizing models?

◯ A: Momentum can cause models to reach the minimum more quickly.

◯ B: Momentum can cause models to reach the minimum more slowly.

◯ C: Given a constant gradient at each time step, the update to the weights will *decrease* over time when using AdaGrad.

◯ D: Given a constant gradient at each time step, the update to the weights will *increase* over time when using SGD without momentum.

SOLUTION:

◯ A: True. Momentum accumulates previous gradients in the update direction, which can help accelerate optimization by maintaining velocity in consistent directions. This helps the model overcome local minima and plateaus, leading to faster convergence to the minimum.

◯ B: True. Momentum can also make the model reach the minimum more slowly as it can overshoot and have to retrace steps. (Note that we accepted either answer here, as the wording may have been confusing)

◯ C: True. In AdaGrad, the update rule divides the learning rate by the square root of the sum of squared gradients (plus). With a constant gradient g, this denominator grows over time as G $= \Sigma g^2$ increases, causing the effective learning rate and thus the weight updates to continuously decrease.

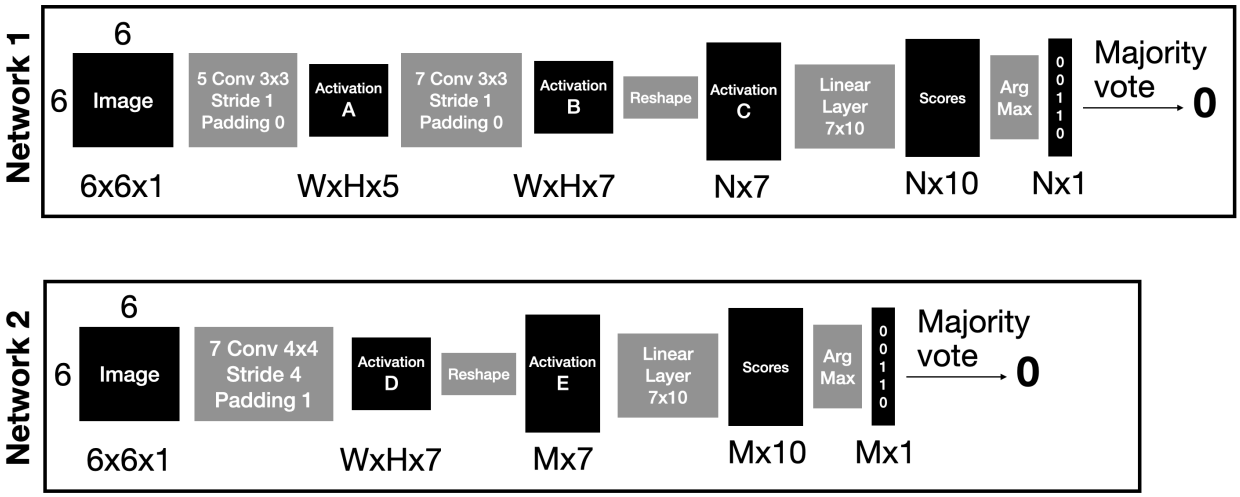
◯ D: False. In standard SGD without momentum, given a constant gradient g, the update will be constant at each time step: = - g. The magnitude of updates neither increases nor decreases over time.

3 Short Answers (8 points) - Recommended 8 Minutes

Please make sure to write your answer only in the provided space.

3.1 Using CNNs to classify numbers

You are using CNNs to classify handwritten numbers as a 0 or 1. However, you decide to do something a little different. Instead of totally flattening out the activations after the CNNs, you decide to classify the letter at each location of the activation map and then take a majority vote over all the classifications. You try this out with two networks which you fully train using cross-entropy loss.

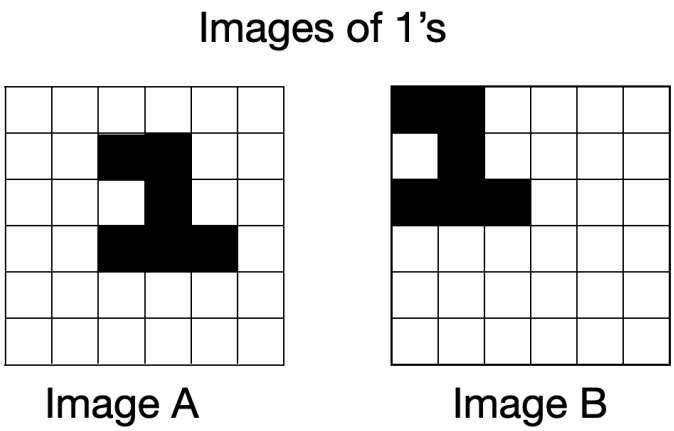


1. What is the shape of Activation A? (2 pnts)

2. What is the shape of Activation C? (2 pnts)

3. What is the shape of Activation E? (2 pnts)

4. You fully train both networks with the same data, such that they have very little train and test error. You can tell from the training curves that neither model is underfit or overfit. You then attempt to classify the two images below. Both networks correctly classify Image A. However, only Network 1 successfully classifies Image B. Network 2 classifies it as a 0, with a 75% vote for 0 and a 25% vote for 1. What about the architecture of Network 2 made this happen? Why is this not the case for Network 1? (2 pnts)



SOLUTION:

1. $4 \times 4 \times 5$
2. 4×7
3. 4×7
4. Network 2 does not get this correct, because it has 1 conv with stride 4. This means that 3 out of the 4 activations don't see any information for image B. This does not happen with network 1 because the lower stride and the multiple layers of convolutions means that each value in activation C has information from a larger region of the image. Also an explanation may include an image like the one below to show the regions of the image which contribute to each region in activation C.